Amendments to the Specification:

Please replace paragraphs [0008] and [0009] with the following amended paragraphs:

[0008] To illustrate, one tool for producing software modeling language SDL code is TELELOGIC TAU ~~Telelogic Tau~~. That tool allows for software programmers to produce a model of the software and the tool produces code from the model. ~~Telelogic Tau~~ TELELOGIC TAU has two potential code generators, ~~C advanced~~ C ADVANCED and C MICRO ~~C micro~~. ~~C advanced~~ C ADVANCED allows a software designer to use all the features of SDL. ~~C micro~~ C MICRO limits the available features with the advantage of producing a smaller code footprint.

[0009] ~~Telelogic Tau~~ TELELOGIC TAU supports both light and tight integration. In light integration, the entire SDL system executes in a single thread. In tight integration, each SDL process becomes a thread. To minimize the code footprint, ~~C micro~~ C MICRO is desirable to be used. However, ~~C micro~~ C MICRO only supports light integration and, accordingly, multiple threads of execution are not supported. Accordingly, it is desirable to have code with a reduced footprint that supports multiple threads of execution.

Please replace paragraph [0016] with the following amended paragraph:

[0016] Figure 5 is an illustration of the structure of SDL ~~C micro~~ C MICRO generated data types.

Please replace paragraph [0036] with the following amended paragraph:

[0036] In the preferred embodiment, an SDL porting layer 110 is used. The preferred SDL porting layer allows for software modeled using a single threaded

code generation tool to run as multi-threaded software. The most preferred application is to code generated using ~~C-micro~~ C MICRO light integration of ~~Telelogic Tau~~ TELELOGIC TAU. The SDL porting layer 110 effectively converts the light integration (single thread) to a tight integration (multiple threads). The most preferred application allows for reduced code footprint (~~C-micro~~ C MICRO, light integration) with the added benefits a multi-threaded environment.

Please replace paragraphs [0038] and [0039] with the following amended paragraphs:

[0038] In other embodiments, the other tools can be used without the SDL porting layer. Additionally, aspects of the SDL porting layer can be used with other single thread code generation tools as well as multi-thread code generation tools other than ~~C-micro~~ C MICRO.

[0039] The basic requirement of the porting layer 110 is to produce the same SDL system behavior as the SDT provided ~~C-micro~~ C MICRO kernel. This includes the ability to support the following SDL model features (SDT-imposed restrictions in parentheses).

SDL Feature:

- Multiple processes
  - Send-Via signals (no output via all)
  - Send-To signals
  - Deferred signals
  - Multiple instances of a process (finite number only)
  - Asterisk input, asterisk state
  - Timers (integer duration only)
  - Signal w/ parameters (no omission of parameters in a signal input)
  - Timer w/ parameter (one integer parameter only)
  - Dynamic process creation (no formal parameters)

- Procedure (no inheritance, no states, no nested procedure call data scope)
- Service
- Predefined Types
- (No RPC)
- (No Service and Priority Input and Output)
- (No Export/Import, View/Reveal)
- (No Enabling condition / Continuous signal)
- (No Any Expression)
- (No Macros)
- (No channel substructure)
- (Names of processes within different blocks must be different)

Please replace paragraph [0041] with the following amended paragraph:

[0041] The options chosen for the preferred code generation are as follows, although others may be used.

- Analyze & generate code (only)
- Code generator: ~~C micro~~ C MICRO
- Prefix: Full
- Separation: Full
- Capitalization: As defined

Please replace paragraph [0049] with the following amended paragraph:

[0049] Figure 7 illustrates the format of a Process Control Block. The data type and its associated table are extensions to ~~C micro~~ C MICRO to support tight integration to the OS API. The process control block 144 has a PID 154, OS handles 156_1 to 156_3 and a process description pointer 158. One OS handle 156_1 points to a message queue. Another OS handle points to a save list 148 and another 156_3 to an active timer list 150. The process description pointer 158 points to a process description 152.